Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

A Stochastic Model of Gene Expression Including Splicing Events

Flávia Alexandra Mendes Penim

Dissertação
Mestrado em Bioinformática e Biologia Computacional
[Bioinformática]

2014

Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

A Stochastic Model of Gene Expression Including Splicing Events

Flávia Alexandra Mendes Penim

Dissertação
Mestrado em Bioinformática e Biologia Computacional
[Bioinformática]

Dissertação Orientada pelo Prof. Doutor Francisco Rodrigues Pinto

2014

“If our ignorance is infinite, the only possible course of action is to muddle through as best we can. (...) Science makes me feel stupid too. It's just that I've gotten used to it. So used to it, in fact, that I actively seek out new opportunities to feel stupid. I wouldn't know what to do without that feeling. I even think it's supposed to be this way.”

- The importance of stupidity in scientific research. J Cell Sci 121, 1771 (June 1, 2008)

Agradecimentos

Muito obrigada. Em primeiro lugar ao Professor Doutor Francisco Rodrigues Pinto, orientador desta tese, pela paciência, disponibilidade e apoio demonstrada ao longo de todo o meu percurso.

Obrigada aos meus pais e irmão por me terem apoioado em cada passo que dei, sempre interessados em saber sobre o meu trabalho independentemente do quão o percebessem ou não.

Obrigada à Natacha pelos serões e apoio, pelos desabafos e partilhas, por todas as discussões construtivas e por todas as outras que nem tanto. Obrigada Bruno, por todas as segundas opiniões e críticas que me levaram a este resultado e por todo o apoio que só tu me podias dar.

Obrigada a todos o que direta ou indiretamente de alguma forma me ajudaram, uns lá mais longe outros aqui mais perto. Do fundo do meu coração, obrigada.

25 de setembro de 2014

Abstract

Proteins carry out the great majority of the catalytic and structural work within an organism. The RNA templates used in their synthesis determines their identity, and this is dictated by which genes are transcribed. Therefore, gene expression is the fundamental determinant of an organism's nature.

The main objective of this thesis was to develop a stochastic computational model able to simulate the gene expression phenomena to best evaluate its critical points including the process of splicing. With this aim, an extended research was performed looking for already described similar models, identifying their strengths and weaknesses, approaches, languages and algorithms used.

Here we present a model developed in Java, which is an object-oriented language or object-oriented programming (OOP) implementing the Gillespie's algorithm. The model receives as input an array of 19 biological parameters. Although at first the model was time consuming and intense, it was optimized reducing the simulation time in approximately 250%.

With this model we wish to take a closer look at the regulation of gene expression, evaluating it with greater accuracy. For that purpose we varied the values of the transcription initiation, RNA degradation and protein degradation rate constants. The results evidenced that the transcription initiation and RNA degradation present the same level of control towards the influence of pre-mRNA, mRNA and protein numbers; in terms of protein numbers, their influence is lower when compared with the protein degradation constant (which has no influence on RNAs numbers).

Resumo

A informação genética de cada indivíduo está guardada no seu *DNA*, uma biomolécula de armazenamento muito estável, de forma a garantir a integridade dos genes de cada um. Contudo sendo muito estável e imutável também torna muito difícil o acesso à informação que contém. É necessário transcrever essa informação para um outro tipo de molécula mais ativa. É este o conceito por detrás da expressão génica.

A expressão génica pode ser resumida em dois passos fundamentais: a transcrição e a tradução. O primeiro é a etapa durante a qual o *RNA* mensageiro (mRNA) é produzido de acordo com o gene que serve como molde. No passo seguinte o mRNA vai servir de base para que o seu conteúdo seja *traduzido* pelos ribossomas para uma nova molécula, a proteína.

As proteínas desempenham a grande maioria das funções catalíticas necessárias a um organismo. O *RNA*, utilizado como base para a sua síntese, determina a sua identidade e esta é ditada pelo gene que anteriormente serviu de molde à sua produção. Assim, não é de admirar que o processo de expressão génica seja tão importante como crítico. Defeitos ou anomalias ao nível da transcrição são responsáveis pela origem de grande parte das doenças que afetam a espécie humana. A maioria dos genes eucariotas (e alguns procariotas) são expressos como um precursor do *mRNA* que é editado e transformado na forma madura do *mRNA*. A este processo dá-se o nome de *splicing*, este passo também essencial é o momento em que porções não codificantes presentes no pré-*mRNA* (intrões de região intragénica) são removidas e as regiões codificantes (exões) são unidas dando origem ao *mRNA*.

A expressão génica é em si um processo denominado estocástico. A palavra 'estocástico' significa 'aleatório' do grego $\sigma\tau\grave{}\chi os$ que significa 'adivinhar'. Os genes são ativados por associações e dissociações aleatórias, a transcrição é tipicamente rara e muitas proteínas são expressas em baixa quantidade por célula. Ao nível celular, a dinâmica química é frequentemente determinada pela ação de apenas um punhado de moléculas, consequentemente flutuações ao nível molecular podem controlar a dinâmica da célula.

Em teoria probabilística um processo estocástico, ou aleatório, é descrito como um conjunto de variáveis aleatórias. Contrariamente ao processo determinístico, ao invés de descrever um processo de uma forma que terá apenas um resultado possível, como por exemplo a resolução de equações diferenciais ordinárias, num processo

estocástico há um certo grau de indeterminação. Mesmo quando o ponto de início da simulação é conhecido há vários caminhos pelos quais a simulação pode evoluir.

De um ponto de vista bioinformático a abordagem por nós seguida, e que culminou no trabalho desta tese, teve como principal objetivo desenvolver um modelo informático de simulação estocástica do processo de transcrição e tradução génica incluindo o fenómeno de *splicing* no processo de maturação dos pré-mRNAs em mRNA maduros. Com o desenvolvimento deste programa é pretendido que seja melhor possível avaliar/identificar os locais de maior impacto e, por consequência, controlo destes, de todo o decorrer da expressão génica.

Neste sentido foi feita uma extensa pesquisa bibliográfica, tentando identificar modelos já existentes reconhecendo tanto as suas mais-valias como as desvantagens, formas de abordagem ao problema, linguagem, algoritmos de iteração usados e conclusões obtidas. Com a informação recolhida foi por nós desenhado um novo modelo.

A linguagem escolhida para o desenvolvimento do programa foi o Java, uma linguagem de programação informática desenvolvida no início dos anos 90. Esta linguagem surgiu como uma resposta à necessidade de uma linguagem mais simples do que a mais popular na altura – C++ – mas tão poderosa como esta. Trata-se de uma linguagem orientada por objetos (do inglês *OOP object-oriented programming*), usa o conceito de objetos que possuem *atributos* que descrevem o seu estado e procedimentos a si associados, os *métodos*, que, medeiam a interação entre objetos alterando o seu estado. É esta interação e os objetos que constituem um programa Java.

No desenho do modelo foi utilizado como algoritmo iterador o algoritmo de Gillespie, publicado pela primeira vez em 1977 por Daniel Gillespie. Nesse artigo descreve a aplicação do algoritmo na simulação de sistemas de reações químicas com um poder computacional limitado. Este algoritmo, também referido como *SSA* (*stochastic simulation algorithm)* pode ser resumido em cinco passos:

1. Inicialização do tempo $t = t_0$ e estado $x = x_0$do sistema;
2. Avaliar todas as reações possíveis $a_j(x)$ e calcular a soma das suas constantes de velocidade $a_0(x)$;
3. Gerar os valores de τ e j;
4. A reação R_j ocorre pela atualização do tempo t e do estado do sistema x;
5. Voltar ao passo 2 (ou terminar a simulação).

O desenho do modelo tem em conta um grande conjunto de parâmetros e constantes biológicas cujos valores foram obtidos de várias publicações científicas referentes a trabalhos experimentais assim como de algumas publicações referentes a outros trabalhos de simulação/modelação. Tendo concluído a fase inicial do desenvolvimento do modelo procedeu-se à sua otimização que resultou numa redução do tempo de execução na ordem dos 250%.

O modelo fornece um output personalizável que permite avaliar a quantidade de pré-mRNA, mRNA e proteína presentes no sistema ao longo do tempo de simulação. Este modelo pode ser aplicado a qualquer gene do código genético humano, variando os valores das constantes biológicas associadas a este.

Da análise do output foi possível avaliar a resposta do modelo face a flutuações no valor dos parâmetros da iniciação da transcrição, degradação do RNA e degradação de proteína. É também possível comparar o grau da intensidade da resposta, pela comparação dos declives das regressões lineares, das simulações com 1 evento de *splicing* por mRNA face às simulações com sete locais.

O modelo estocástico da expressão génica apresentado, embora seja relativamente intenso, permite a execução de simulações com aproximadamente 50 milhões de iterações correspondendo a um tempo de simulação de 15.000 segundos num período de tempo inferior a 120 minutos, havendo disponibilidade de processadores podem ser corridas várias simulações em simultâneo. Uma das vantagens da linguagem escolhida é a versatilidade que lhe permite que o modelo seja executado em qualquer sistema operativo.

A inclusão dos eventos de *splicing* num modelo estocástico permitem dar um novo olhar sobre a análise do processo de expressão génica e a sua regulação.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The great advances in genetic and molecular biology, occurred in the past years, have brought an unprecedented flood of genomic data. The need to analyse this data – to understand how genes and proteins work together – has created a new challenge, which led to a significant increase in the use of computer power. The aim of bioinformatics is to develop both tools and methods that will allow the interpretation of this data, as well as create models that mimic biological phenomena, against which data or patterns can be compared (Roussel, 2006).

Today's technology enables scientists to take a closer look to complex phenomena and examine the underlying mechanisms and interactions. One good example is the process of gene expression that is now being observed in single-cell and single-molecule experiments (Yu, 2006).

1.1 Gene Expression

What does it mean to 'express' a gene? Genetic information is stored as DNA. DNA is a very stable storage molecule, which is important because genetic information must be conserved from one generation to another. However, it is also a very inactive molecule. For the information stored within the DNA to be utilized then, it needs to be converted into another, more active, molecular form. This is the concept behind 'expressing' a gene (White, 2001).

Two major steps can briefly describe gene expression: transcription and translation. The first one, transcription, is the stage during which messenger RNA (mRNA) is produced according with the information present in the DNA. In the translation process that follows, the mRNA serves as a template that ribosomes use to synthetize proteins (*Figure 1*).

Proteins carry out the great majority of the catalytic and structural work within an organism. The RNA templates used in their synthesis determines their identity, and this is dictated by which genes are transcribed. Therefore, gene expression is the

fundamental determinant of an organism's nature. Thus, it is no surprise that defects in transcription are known to characterise the majority of human diseases (White, 2001) (Keren, Lev-Maor, & Ast, 2010).

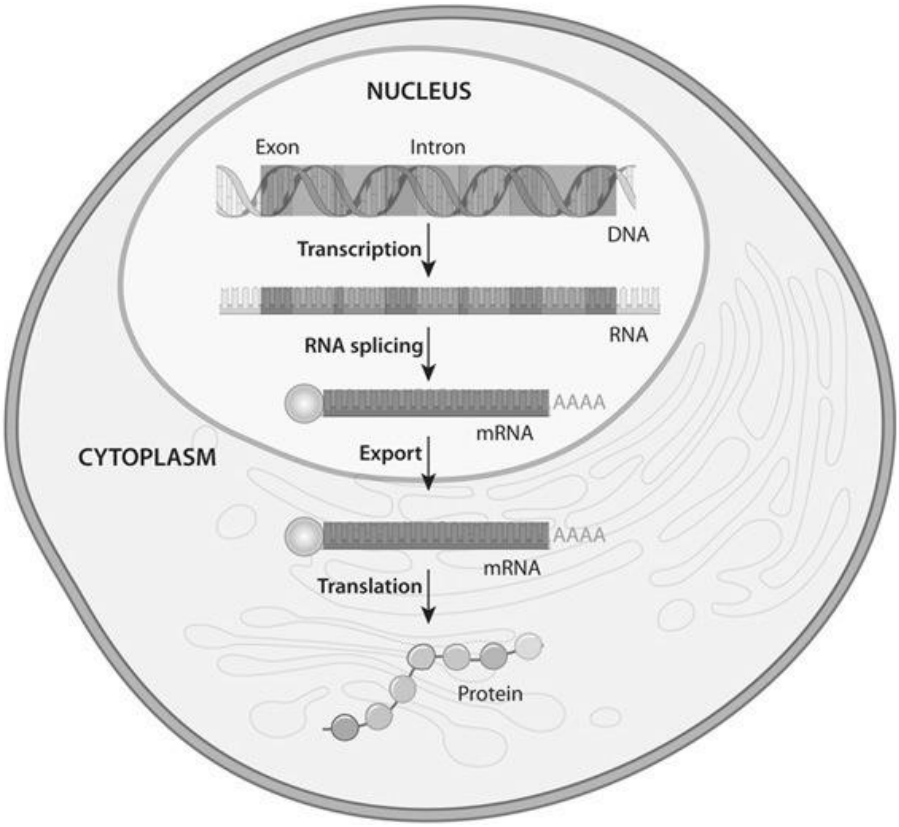

Figure 1: Gene expression. Both coding and noncoding regions of DNA are transcribed into mRNA. Some regions are then removed (introns) during initial mRNA processing. The remaining exons are then spliced together, and the spliced mRNA molecule is prepared for export. Once in the cytoplasm, the mRNA can be used to construct a protein. *In Scitable, Gene Expression © Nature Education 2014.*

RNA transcription is a vital event in gene expression and is strongly controlled during all of the stages of synthesis of the transcript, including initiation, elongation, and termination. An error in the regulation during any of these processes can result in aberrant gene expression, compromise survival of single celled creatures, and lead to disease in higher organisms (Greive, et al., 2008). Due to the fact that protein numbers follow RNA numbers, small oscillations in RNA numbers are likely to have downstream effects on the phenotype (Potapov, Mäkela, Yli-Harja, & Ribeiro, 2012).

1.2 Splicing and Alternative Splicing

Most eukaryotic genes (and some prokaryotic ones) are expressed as precursor mRNAs (pre-mRNAs) that are converted to mRNA by splicing, an essential step of gene expression in which noncoding sequences are removed and coding sequences are ligated together (Will & Lührmann, 2011; Keren, Lev-Maor, & Ast, 2010). The process

called RNA splicing comprises the removal of certain sequences named introns, the word derives from the term *intragenic region*, meaning a region inside a gene. The final mature mRNA consists of the remaining sequences, called exons, which are linked to one another through the splicing process (Clancy, 2008).

But how does exactly 'splicing occurs'? The biochemical mechanism by which splicing occurs has been studied in a number of systems and is now fairly well characterized. Splicing is catalysed by the spliceosome, a large RNA-protein complex composed of five small nuclear ribonucleoproteins (snRNPs). Introns are removed from primary transcripts by cleavage at conserved sequences called splice sites. The event of splicing consists of a two-step biochemical process, both steps involving transesterification reactions that occur between RNA nucleotides. The splice sites are found at the 5' and 3' ends of introns. Most eukaryotic introns for genes transcribed by RNA polymerase II (RNApol) begin with the bases GU and end with the bases AG. In the first step the ester bond between the 5' phosphate of the intron and the 3' oxygen of the upstream exon is exchanged for an ester bond between the 5' phosphate of the intron and the 2' oxygen of the branch point A residue. In the second reaction, the ester bond between the 5' phosphate of the downstream exon and the 3' oxygen of the intron is exchanged for an ester bond between the 5' phosphate of the downstream exon and the free 3' oxygen of the upstream exon (Figure 2). This results in the connection of the upstream exon to the downstream one and the release of the intron (Latchman, 2010).

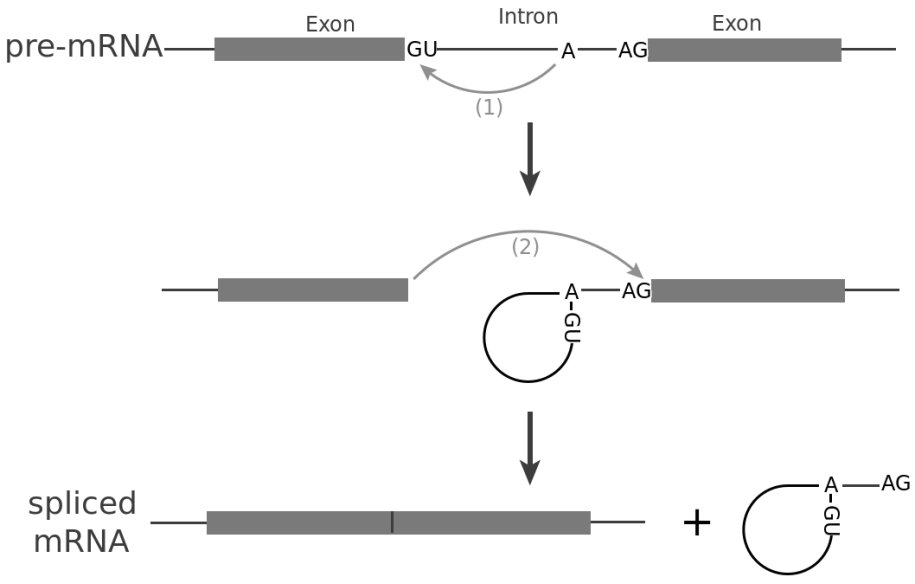

Figure 2: Biochemical mechanism of splicing. "RNA splicing reaction" by BCSteve - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons.

Alternative splicing has been described in numerous cases, both in mammals and other organisms. Cases of alternative RNA processing occur in the genes involved in a wide variety of different cellular processes. Indeed, recent studies of the whole human genome concluded that over 90% of human genes with multiple exons are alternatively spliced (Wang, 2010). Alternative splicing is frequently constitutive, leading to the same protein outcomes under different conditions. However, there are also many cases in which the splicing pattern is regulated in order to change the genetic result according to the cell type circumstances (White, 2001).

The different combinations of exons that are included, or not, in the mRNA leads to the origin of different isoforms from one single gene (*Figure 3*) and is the foundation for the discrepancy between the estimated 24,000 protein-coding genes in the human genome and the 100,000 different proteins that are hypothesized to be produced (Keren, Lev-Maor, & Ast, 2010).

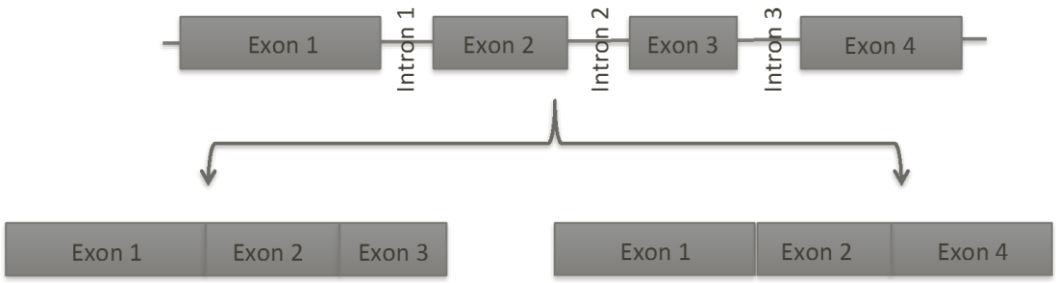

Figure 3: Alternative splicing.

1.3 Stochastic Process

There are two major approaches for mathematically describing the time behavior of chemical reactions: *deterministic* models, which are based on differential equations regarding the time evolution as a continuous, wholly predictable process and *stochastic* simulations, regarding the time evolution as a kind of random-walk process which is governed by a single differential-difference equation (the "master equation") (Gillespie, 1977).

In probability theory, a stochastic process, or random process is a collection of random variables. Contrarily to a deterministic process, instead of describing a process that can only develop to one outcome (as for instance the solution of ordinary differential equations) in a stochastic process there is some indeterminacy, even when the starting point is known, there are many directions in which the process can evolve.

The word ‘stochastic’, meaning ‘random’, comes from the Greek $\sigma\tau\grave{\chi}os$ meaning ‘to guess’. Mathematically a random process is described as being a collection of random values where t is a parameter that runs over an indexed set T (Stirzaker, 2005).

$$\{X(t): t \in T\} \quad (1)$$

Usually the t is called *time-parameter* or time, and $T \subseteq \mathbb{R}$. Each $X(t)$ takes values in some set $S \subseteq \mathbb{R}$ called the *state space*. The $X(t)$ represents the state of the process at time t. For example $X(t)$ may be the number of persons in a waiting line at a given time *t*, the maximum temperature on a day *t*, (…).

1.4 Stochasticity in Gene Expression

In general, a chemical reaction occurs when two or more molecules of suitable kinds collide in a proper way. The "stochastic formulation" of chemical kinetics is simply a consequence of taking seriously the fact that collisions in a system of molecules occur in an essentially random manner (Gillespie, Exact Stochastic Simulation of Coupled Chemical Reactions, 1977).

Gene expression is an inherently stochastic process: Genes are activated and inactivated by random association and dissociation events, transcription is typically rare, and many proteins are present in low numbers per cell (Gillespie, Exact Stochastic Simulation of Coupled Chemical Reactions, 1977). At the level of the cell, the chemical dynamics are frequently determined by the action of only a few molecules and, consequently, molecular fluctuations may dominate the dynamics (Rao & Arkin, 2003).

1.5 Gillespie Algorithm

The *Gillespie Algorithm* was first published in 1977 by Daniel Gillespie. In this paper he describes the usage of this algorithm in order to accurately simulate systems of chemical reactions with limited computational power. Gillespie’s Stochastic Simulation Algorithm (SSA) is fundamentally a precise procedure for numerically simulating the time evolution of a well-stirred chemical reacting system by taking proper account of the randomness present in such system (Gillespie, Exact Stochastic Simulation of Coupled Chemical Reactions, 1977).

In a stochastic formulation we consider a well-mixed system of N chemical species $\{S_1, \dots, S_N\}$ interacting through M chemical reactions$\{R_1, \dots, R_M\}$. With $X_i(t)$

representing the number of molecules of species S_i in the system at the time t, we wish to study the evolution of the vector $X(t) = \big(X_i(t), \dots, X_N(t)\big)$. Given that initially $X(t_0) = x_o$ is possible to deduce a time evolution equation for this function using the laws of probability. However, the equation obtained is rarely of much use in computing. With this aim, a new function as been described by Gillespie $p(\tau, j|x, t)$ (Gillespie, 1976). The function is defined so that $p(\tau, j|x, t)d\tau$ is the probability, given $X(t) = x$, that the next reaction in the system will occur in the infinitesimal time interval $[t + \tau, t + \tau + d\tau]$, and will be a R_j reaction. Is possible to summarize this function as the combined probability density function of the two random variables: τ – *time to the next reaction* and j – *index of the next reaction* and where a_j stands for the value of the kinetic parameter associated with the reaction j . An analytical expression for $p(\tau, j|x, t)$ is possible to deduce and with the needed rearrangements the solution is found to be $P_0(\tau|x, t) = \exp(-a_0(x)\tau)$, where:

$$a_0(x) \equiv \sum_{j'=1}^{M} a_{j'}(x) \qquad (2)$$

Therefore we can condense the *stochastic simulation algorithm* (SSA) into the following steps (Gillespie, 1976; 1977):

6. Initialize the system's time $t = t_0$ and state $x = x_0$;
7. Evaluate all the $a_j(x)$ and their sum $a_0(x)$;
8. Generate values for τ and j;
9. The reaction R_j occurs by updating the time t and the system state x;
10. Return to step 2 (or end the simulation).

1.6 Motivation

Due to the importance of a correct control of gene expression the objective of this thesis is to develop an informatics model that mimics the process of transcription and translation. This approach brings an improvement from other models already existent – *splicing events*. This model simulates the path from the gene until the protein including the process of splicing (with varying number of splicing sites).

It's at the level of pre-mRNA editing that the model presented here makes the difference. With this, we aim to better understand the molecular impact of splicing and locate the more effective points of gene expression control.

Chapter 2

Model Description

In this chapter we will provide a detailed description of the model developed and discuss step by step the simulation flow. The approach to the problem will be explained along with the model course "from the gene to the protein".

2.1 Programming Language – JAVA

Java is a computer programming language developed during the early 90's and first released in 1995 by a team lead by James Gosling for Sun Microsystems. It has evolved over the years and is today a very successful language with over 6.5 million developers worldwide (Oracle Corporation, 2014).

Why choose Java? The language was developed with a few key principles in mind: *Ease of use* – The founding ideas of Java came from another programming language named C++, although still used, C++ was felt to be very complex so Java improved its usability while providing a new language as powerful as easy to use (Oracle Corporation, 2014).

It is an object-oriented language or object-oriented programming (OOP) which uses the concept of "objects" that have data fields, *attributes* that describe the object, and associated procedures – *methods*. Objects interact with each other to design applications and computer programs. But what exactly are objects? Objects are the key to understanding object-oriented technology, looking around you can find many examples: your car, dog, desk and in this case a gene, RNApol, pre-mRNA, spliceosome, mRNA, ribosome and protein.

Real-world objects share two characteristics: They all have state and behaviour. Cats have state (name, colour, breed …) and behaviour (sleeping, eating …). Tables also have state (colour, number of legs, wood type …) and behaviour (cleaning, oiling, removing scratches …). Finding the state and behaviour for real-world objects is a great way to begin thinking in terms of OOP (Oracle Corporation, 2014).

An object keeps its state in *fields* (variables in some programming languages) and exposes its behaviour through *methods* (functions in some programming languages).

Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication. Hiding internal state and requiring all interaction to be performed through an object's methods is known as *data encapsulation* – a fundamental principle of object-oriented programming (*Figure 4*). Let's look at a car as an example:

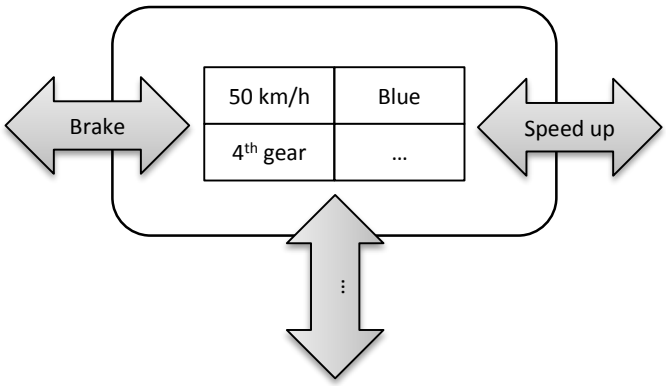

Figure 4: A car modelled as a software object. Adapted from Oracle Java Documentation – *What is an object?*

By attributing state and providing methods for changing that state, the object remains in control of how the outside world is allowed to use it. For example, if the car only has 5 gears, a method to change gears could reject any value that is less than 1 or greater than 5.

2.1.1 Primitive Data Types

In JAVA you can encounter several different primitive data types, in total eight, which are:

i. byte;
ii. short;
iii. int;
iv. long;
v. double;
vi. float;
vii. boolean;
viii. char.

In addition to these eight, Java also provides support to character strings. The String class is not technically a primitive data type, but considering the special support given to it by the language, we'll probably tend to think of it as such (Oracle Corporation, 2014).

2.1.2 Classes

In our daily life we often find many individual objects all of the same kind. There may be thousands of other cars in existence, all of the same make and model. Each car was built from the same set of blueprints and therefore contains the same components. In object-oriented terms, we say that your car is an instance of the class of objects known as cars. A class is the blueprint from which individual objects are created.

Bellow we show a class declaration example (*Table 1*). The *class body* (the area between the braces) contains all the necessary code for the life cycle of the *objects* created from the class: *constructors* for initializing new objects, declarations for the *fields* that provide the state of the class and its objects, and *methods* to implement the behaviour of the *class* and its *objects*.

Table 1: Class declaration example.

```
class MyClass {
      //field, constructor and
      //method declarations
}
```

Every class is different from the previous, but every single one will be organized accordingly the same basic principles we have just described (Oracle Corporation, 2014).

2.2 Organization

In order to approach the "problem" the first step is to design the model. To do that we must look back at the flow of events and participants of gene expression (*Figure 5*). That can be achieved by reducing it to the following stages:

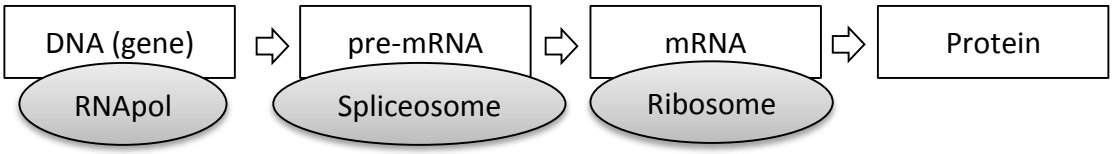

Figure 5: Steps and participants of the gene expression process.

With this representation we can divide the participants into two categories – active and passive – being the active participants the RNApol, Spliceosome and Ribosome, and the passive ones the gene, pre-mRNA, mRNA and protein. In other words what this means in JAVA is that some objects will act upon others and change

their state. Each one of the objects described has actions associated to them, the so called *methods* (Oracle Corporation, 2014).

2.2.1 Model Objects

Gene – Looking back towards biology, what does characterize a gene? A gene is composed by three main regions – promoter (region of DNA close to the transcription start site which directs transcription), coding region and stop codon (Latchman, 2010). It can vary in number of base pairs (bp). Translating this to JAVA (*Table 2*), it means the gene object will have a dimension, this is a value that cannot be changed during the course of the simulation and it's a final value. The promoter will be represented as a parameter with two values – true for available and false for occupied – stating whether the promoter is available for transcription to start or not (if a RNApol is already transcribing the gene, another will only be able to start transcribing when the first progresses enough along the gene). The gene will also have, in the form of a list, a record of all the RNApol that are currently attached to it.

Table 2: Gene parameters scheme.

Gene
promotorDisp: boolean = true dimGene: int RNApolLink: ArrayList<> = ArrayList<RNApol>

RNApol – This enzyme connects to the gene promoter region and starts the transcription process producing the pre-mRNA (Epshtein & Nudler, 2003). For this process to be modelled there are several parameters to consider (*Table 3*). Those parameters are: the RNApol dimension (a final value) determining the space it will occupy along the gene during transcription (two or more RNApol cannot occupy the same space on a gene) this refers not only to the nucleotide currently being transcribed, but also to the physical space a RNApol has (Greive & Hippel, Thinking quantitatively about transcriptional regulation, 2005); if it's currently transcribing or not; the transcription rate (how fast does the RNApol progress along the gene); the position of the RNApol (what base is it transcribing at a given simulation time); the gene it's transcribing; the pre-mRNA it is currently synthetizing; all the pre-mRNA that it has already transcribed and have not been either degraded or "edited" into mature mRNA's; the initiation constant; the transcription rate; and the abort constant.

Table 3: RNApol parameters scheme.

RNApol
velTrans: int posRNApol: int dimRNApol: int link: boolean = false current: Gene g: Gene currentP: premRNA pmRNA: arrayList<> = arrayList<premRNA> kl: double ka: double kd: double

pre-mRNA – The precursor messenger RNA is an immature single strand of mRNA. It exist only briefly before being completely processed into the mature mRNA. Pre-mRNA's are composed by two different types of segments – exons and introns (Clancy, 2008). Exons are the segments kept during the event of *splicing* where the introns are removed (Keren, Lev-Maor, & Ast, 2010). Therefor pre-mRNAs can be described by (*Table 4*): the number of splicing sites (total, available for splicing and done); if it is currently connected to a spliceosome; which spliceosome is it connected to; if it's still in synthesis (connected to a RNApol) ; if it has been fully processed and transported from the cell nucleous to the cytosol.

Table 4: Pre-mRNA parameters scheme.

pre-mRNA
splSitesTT: int splSitesA: int splSitesD: int cytosol: boolean = false spliceosome: boolean = false spl: Spliceosome rnaP: boolean = true

Spliceosome - It is a large and complex molecular machine found primarily within the nucleous of eukaryotes, it removes introns from a transcribed pre-mRNA. This process is generally referred to as splicing. Only eukaryotes have spliceosomes (Will & Lührmann, 2011). The spliceosome characteristics needed to this model are (*Table 5*): if it is currently "editing" any pre-mRNA; if it is, which pre-mRNA is it; constant of velocity for the gathering of all the splicing complex elements; constant for the splicing event; constant for the termination and transport of the new mRNA to the cytosol.

Table 5: Spliceosome parameters scheme.

Spliceosome
link: boolean = false currentP: premRNA arrayM: ArrayList<> = ArrayList<mRNA> kl: double ks: double kt: double

mRNA – The messenger RNA is the mature form of the transcript originated by transcription. It has no introns and exists in the cell cytoplasm where it will be used as the model for the protein production by ribosomes (Nelson & Cox, 2008). A single mRNA may be translated simultaneously by multiple ribosomes without them overlapping each other. It's characterized by its dimension; if it's available for a ribosome to connect and a list containing all the ribosomes currently translating it (*Table 6*).

Table 6: mRNA parameters scheme.

mRNA
dim: int avail: boolean = true RibosomeLink: ArrayList<> = ArrayList<Ribosome>

Ribosome – It is a large and complex molecular engine, found within all living cells, that serves as the primary site of biological protein synthesis (translation). Ribosomes link amino acids together in the order specified by mRNA molecules (Clancy, 2008). The ribosomes and associated molecules are also known as the *translational apparatus*. The parameters that characterize the ribosome are: translational velocity; its position on the mRNA; its dimension (Firczuk, et al., 2013); if it's currently translating; if it is translating, which molecule of mRNA is it connected to and the protein in synthesis; list of all the proteins it has already synthesised; constant of initiation, progress and abortion (*Table 7*).

Table 7: Ribosome parameters scheme.

Ribosome
velTrans: int posRib: int dimRib: int link: boolean = false current: mRNA currentProt: Proteina ArrayPr: ArrayList<> = ArrayList<Protein> Kl: double Ka: double Kd: double

Proteins – Large biological molecules, or macromolecules, consisting of one or more long chains of amino acid residues. Proteins perform a vast array of functions within living organisms. A linear chain of amino acid residues is called a polypeptide. A protein contains at least one long polypeptide. Short polypeptides, containing less than about 20-30 residues, are rarely considered to be proteins and are commonly called peptides. Once formed, proteins only exist for a certain period of time and are then degraded. Like other biological macromolecules such as polysaccharides and nucleic acids, proteins are essential parts of organisms and participate in virtually every process within cells (Nelson & Cox, 2008).

As the final product of the intended simulation, they are characterized by one single parameter, which is if they are completely synthesised or not (*Table 8*).

Table 8: Protein parameters scheme.

Protein
Complete: boolean = false

2.2.2 Methods

Associated with each one of the objects described above exist *methods* that will act as behaviour changers, mediating the interaction between objects.

The gene itself is an object with no possible action on itself or other object, it will remain the same during the simulation course. On the other hand RNApol can act upon the gene, transcribing it into a pre-mRNA and ending the transcription. This summarizes the three main actions that each RNApol can execute and that are modelled (*Table 9*).

Table 9: RNApol methods.

RNApol
Link(Gene) Progress() Disconnect ()

The pre-mRNA obtained is a molecule that will undergo changes operated by other, the spliceosome, the *state* of the object that represents pre-mRNA will suffer changes made by the *methods* of the spliceosome that represent its actions. The splicing complex needs to be gathered (Will & Lührmann, 2011), which is represented by the method “Link”. After being connected to the pre-mRNA if there is any splicing site available, splicing may occur through the method “Splicing”. When all the splicing sites have been edited, a mature mRNA becomes available for transport from the nucleous to the cytosol, this is represented by the “Transport” method (*Table 10*).

Table 10: Spliceosome methods.

Spliceosome
Link(premRNA): boolean Splicing() Transport()

The mature molecule of mRNA does not act or change others, therefore it has no methods associated with it. It will be translated by the ribosome which is the "active" object in the translation representation.

The ribosome is represented similarly to the RNApol, it first action is to connect to a mRNA, then slide along it and disconnect from it, when it reaches the end of the sequence or in case it aborts the process, degrading the incomplete protein (*Table 11*). This is represented in Java by three main *methods*:

Table 11: Ribosome methods.

Ribosome
Link(mRNA) Progress() Disconnect()

The degradations also have to be in the model, but to simplify the model, the degradation of pre-mRNA, mRNA and protein were added as methods of the RNApol, spliceosome and ribosome respectively. Although in reality the event of degradation is not performed by any of them, this is just a way of organizing the model without the need to create an extra object.

2.2.3 Other Methods and Classes

Beside all the methods and classes previously described, that are the representation of biological elements present in the process of gene expression, it was necessary to add a set of extra methods to each object as well as three other classes. This methods and classes do not represent any biological element, but are needed in order to properly organize the model, as well as to allow the model to work correctly.

This extra methods and classes intervene in crucial steps as the output data retrieval and collect, in the algorithm implementation and model and variables initialization. The final organization of the model can be seen in the *Appendix*.

2.2.4 Output

During the simulation time and at regular intervals the *state* of the model is recorded into a text file. The data collected is then used to build a plot that provides visual information about the behaviour of the species in study (pre-mRNA, mRNA and

protein). Beside the visual information, with the raw data is also possible to do the necessary statistical analysis.

2.3 Algorithm Implementation

In order to simulate the model dynamics we need to implement the already described, Gillespie Algorithm (*page 6*).

To initialize the system, the user must give to the program an array of strings containing all the needed parameters values (*Table 12*). These parameters will then be converted by the program into their correct data types (*int*, *boolean*, *double*…). Each parameter value has a specific place in the array and its order cannot be changed. The parameters in their correct order are:

Table 12: Description of the parameters necessary to initialize the simulation.

Order	Parameter	Unit	Data Type
[0]	Simulation time	s	Double
[1]	Saving step	iter	Integer
[2]	Gene dimension	nuc	Integer
[3]	Number of RNApol	-	Integer
[4]	Number of spliceosomes	-	Integer
[5]	Number of ribosomes	-	Integer
[6]	Splicing site number	-	Integer
[7]	RNApol link constant	s^{-1}	Double
[8]	RNApol progress constant	s^{-1}	Double
[9]	RNApol disconnect constant	s^{-1}	Double
[10]	Pre-mRNA degradation constant	s^{-1}	Double
[11]	Spliceosome link constant	s^{-1}	Double
[12]	Spliceosome progress constant	s^{-1}	Double
[13]	Spliceosome transport constant	s^{-1}	Double
[14]	mRNA degradation constant	s^{-1}	Double
[15]	Ribosome link constant	s^{-1}	Double
[16]	Ribosome progress constant	s^{-1}	Double
[17]	Ribosome disconnect constant	s^{-1}	Double
[18]	Protein degradation constant	s^{-1}	Double
[19]	Saving file name	-	String

Given the parameters and with the system initialized we need to evaluate all $a_j(x)$, which represent all the possible *actions* in the system at its current state. Each reaction has a velocity constant associated with (k_β), this constant represents the likelihood of a reaction to occur. The sum $a_0(x)$ is the sum of all the constants of the reactions that can occur. The reactions that cannot occur will have a constant value equal to zero.

In our system exist 13 possible reactions that will be available to occur depending on their reagents existence. The reactions are:

$$Pro + RNApol \xrightarrow{k_l} Pro.RNApol \quad (3)$$

$$Pro.RNApol \xrightarrow{k_p} Pro + gene.RNApol.premRNA \quad (4)$$

$$gene^p.RNApol.premRNA \xrightarrow{k_p} gene^{p+n}.RNApol.premRNA \quad (5)$$

$$gene^p.RNApol.premRNA \xrightarrow{k_d} gene + RNApol + premRNA \quad (6)$$

$$premRNA \xrightarrow{k_{dg}} \quad (7)$$

$$premRNA + Spl \xrightarrow{k_l} premRNA^0.Spl \quad (8)$$

$$premRNA^n.Spl \xrightarrow{k_s} premRNA^{n+1}.Spl \quad (9)$$

$$premRNA^n.Spl \xrightarrow{k_t} mRNA + Spl \quad (10)$$

$$mRNA \xrightarrow{k_{dg}} \quad (11)$$

$$mRNA + Rib \xrightarrow{k_l} mRNA.Rib \quad (12)$$

$$mRNA^p.Rib \xrightarrow{k_p} mRNA^{p+n}.Rib.Protein \quad (13)$$

$$mRNA^p.Rib.Protein \xrightarrow{k_d} mRNA + Rib + Protein \quad (14)$$

$$Protein \xrightarrow{k_{dg}} \quad (15)$$

Depending on the reaction, the associated constant may need be multiplied by another value (number of mRNAs, number of splicing sites...). These reactions are: *Reaction 7* – pre-mRNA degradation, the constant of degradation is multiplied by the number of existent pre-mRNA; *Reaction 8* – spliceosome connection to a pre-mRNA, the constant is multiplied by the number of existent pre-mRNA; *Reaction 9* – splicing, the constant of splicing is multiplied by the number of available splicing sites; *Reaction 11* – mRNA degradation, the constant of degradation is multiplied by the number of existent mRNA; *Reaction 12* – ribosome connection, the constant is multiplied by the

number of existent mRNA; *Reaction 15* – protein degradation, the constant of degradation is multiplied by the number of existent proteins in the system.

At the very first step the only possible action is to one of the available RNApol to connect to the gene promoter region (reaction 3). As the process is just starting there isn't yet any other species (pre-mRNA, mRNA and protein). With the simulation evolution other species and reactions will became available, to have this in account we need to verify from all the 13 reactions which are the ones able to happen.

The number obtained by the sum of the velocity constants (which in some cases are multiplied by the number or reangents present) is used to generate a random number from an exponential function that will represent the time step of increment in the model (τ). Each step of the model, depending on the reactions possible to occur, will have a different time increment.

Simultaneously a second random number *j* is generated that will range from 0 to 1. This number will be used in the raffle of which will be the next reaction to take place in the simulation. The raffle is executed in following order:

1. What element? (RNApol, Spliceosome or Ribosome);
2. Which one of the available element (if a Ribosome was chosen, which one of the n ribosomes);
3. Which reaction (link, progress, disconnect or degrade).

As previously referred, although the RNApol, spliceosome and ribosome are not involved in the pre-mRNA, mRNA and protein degradation, this reaction has been associated to them only in order to facilitate the raffle execution.

In the following figure it's possible to better visualize the three steps in which the raffle is organized. It is highlighted an example where the chosen reaction is one of $RNApol_4$.

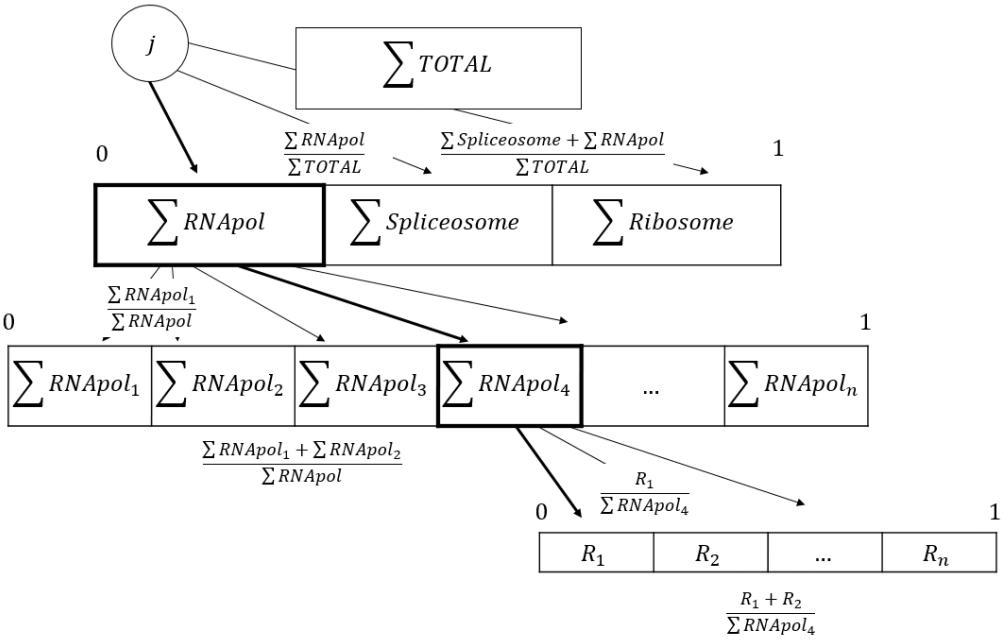

Figure 6: Raffle execution organization.

2.4 Planning

Of the whole process of transcription and translation there are three steps that are potentially the more influential and easy to manipulate. These three steps are the transcription initiation, the RNA degradation and protein degradation. The values of the associated constants (a_j) to this parameters were varied from its central value (100%) to +/- 10%. Two sets of simulations were executed, one with only one splicing site versus another with seven sites. Each simulation replicated a period of 15.000s, this was the time chosen in order that all the species reached their *steady state*. Being this a stochastic simulation, the mentioned *steady state* does not reach a constant value, but rather a constant *moving average*. Each simulation was replicated 4 times (*Table 13*). To evaluate the *steady state* value of each one of the three species we calculated the average value with the last 14.000s of simulation for the pre-mRNA and mRNA and with the last 9.000s for the proteins.

With the data obtained from each set of variations of the constants we did a linear regression whose dependent variables are the pre-mRNA, mRNA and protein steady state numbers (normalized by the steady state number with the central parameter value) and the independent variable is the relative value of the parameter (90%, 100% and 110%). With this relative scale the adjusted line slope represents a normalized sensitivity that quantifies the degree of influence of the parameter on the *steady state* number of each species (pre-mRNA, mRNA and protein) being comparable between

each other. All the linear regressions were calculated using the excel function *linest*, and confidence intervals of 95% were determined for each slope.

The number of ribosomes available for a gene also takes a significant role in this process. Being an essential "worker", a reduced number of ribosomes available will represent a "bottle neck" in the overall system flow. The simulations were performed with 100 ribossomes, because exploratory simulations with fewer ribossomes introduced the referred bottleneck.

Table 13: Explanation of the organization of the number of simulations performed. Each simulation was repeated 4 times (replicates) totalizing a number of 72 simulations.

	RNApol k_l			RNA k_{dg}			Protein k_{dg}			Splicing Sites
100 Ribosomes	90%	100%	110%	90%	100%	110%	90%	100%	110%	1
	90%	100%	110%	90%	100%	110%	90%	100%	110%	7

2.5 Optimization

In the final stage of development of the model and after several tests we came to the conclusion that the program was far too slow and costly in terms of processing CPU capacity. Therefore we proceed to an examination of the program code[1] in order to find what could be changed to optimize the model. Two major changes were made: first, the list containing all the completely synthetized protein, whose dimension could get to values of thousands, was partially converted into an *integer*. This change could be done due to the fact that the protein object (when complete) lacks parameter fields. Second, the protein degradation was changed to an event occurring after each simulation step (*Reaction 16)*, by the direct calculous of the predicted number of protein (np) degraded in the time interval ($\Delta\tau$) obtained in the raffle by a Poisson distribution (*Equation 16*), λ represents the Poisson parameter calculated by $\lambda = \Delta\tau \times k_{pd} \times np$. This two changes allowed a great decrease of the simulation time.

$$f(k,\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \qquad (16)$$

[1] The complete and final version of the code can be found in the attached CD.

Chapter 3

Results e Discussion

In this chapter we will present a set of results of simulations from the model we've been describing. To proceed with the simulation it was necessary to know the values of the 19 parameters the model receives. These values were retrieved from several papers and, has was expected, not all of them in agreement. All the values used and their source can be found in *Table 14*.

Table 14: Values and references of the parameters used in the first set of simulations. The values are mainly obtained from *E. coli* measurements. *Iter* stands for iterations and *bp* for base pairs.

Order	Parameter	Value	Unit	Ref
[0]	Simulation time	15.000	s	-
[1]	Saving step	10.000	iter	-
[2]	Gene dimension	2.000	bp	-
[3]	Number of RNApol	40	-	(Zhu, Ribeiro, Salahub, & Kauffman, 2007)
[4]	Number of spliceosomes	20	-	-
[5]	Number of ribosomes	40 - 100	-	(Krebs & Lewin, 2011)
[6]	Splicing site number	1 - 7	-	-
[7]	RNApol link constant	0,0245	s^{-1}	(Potapov, Mäkela, Yli-Harja, & Ribeiro, 2012)
[8]	RNApol progress constant	65	s^{-1}	(Greive & Hippel, 2005) (Epshtein & Nudler, 2003)
[9]	RNApol disconnect constant	0,00019	s^{-1}	(Potapov, Mäkela, Yli-Harja, & Ribeiro, 2012)
[10]	Pre-mRNA degradation constant	0,025	s^{-1}	(Potapov, Mäkela, Yli-Harja, & Ribeiro, 2012)
[11]	Spliceosome link constant	56,2	s^{-1}	(Huranová, et al., 2010)
[12]	Spliceosome splicing constant	125,6	s^{-1}	(Singh & Padgett, 2009)
[13]	Spliceosome transport constant	533	s^{-1}	(Grünwald, Singer, & Rout, 2011)
[14]	mRNA degradation constant	0,025	s^{-1}	(Potapov, Mäkela, Yli-Harja, & Ribeiro, 2012)
[15]	Ribosome link constant	0,4	s^{-1}	(Mäkelä, Lloyde-Price, Yli-Harja, & Ribeiro, 2011)
[16]	Ribosome progress constant	1000	s^{-1}	(Ribeiro, Häkkinen, & Lloyd-Price, 2012)

[17]	Ribosome disconnect constant	0,000114	s^{-1}	(Potapov, Mäkela, Yli-Harja, & Ribeiro, 2012)
[18]	Protein degradation constant	0,0017	s^{-1}	(Ribeiro, Häkkinen, & Lloyd-Price, 2012)
[19]	Saving file name	-	-	-

Each simulation has a duration of 15.000s and translates the time evolution during that period of the pre-mRNA, mRNA and protein numbers (*Figure 7*).

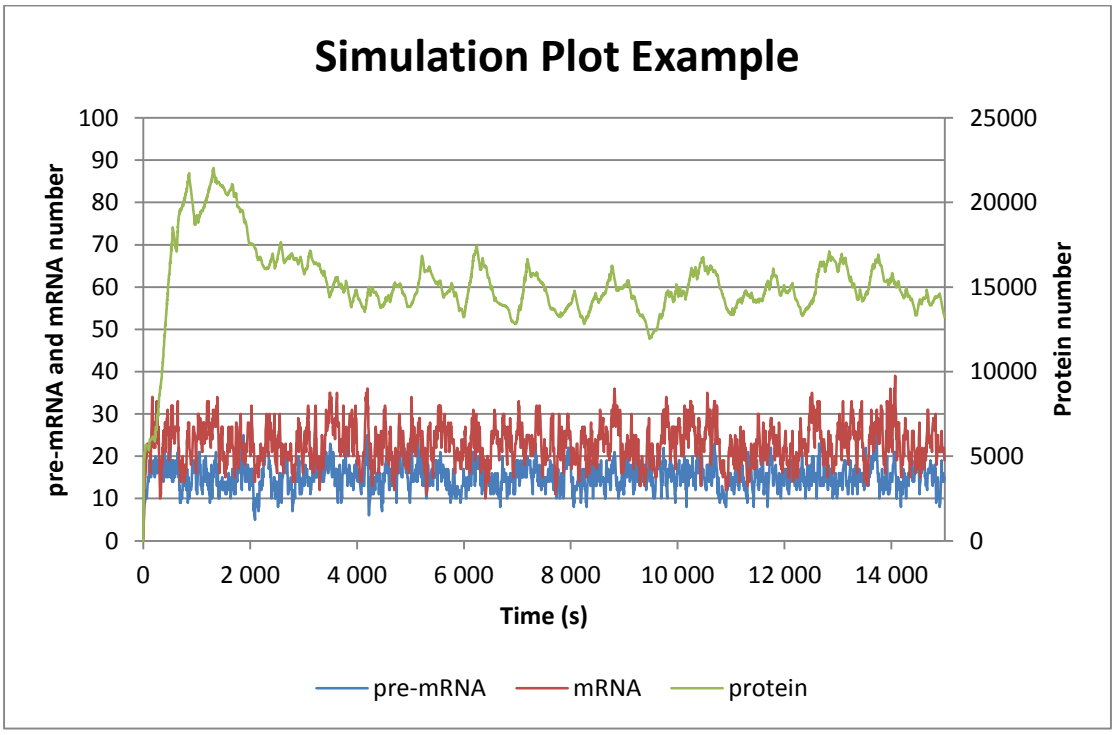

Figure 7: Output plot example. Values of pre-mRNA and mRNA scaled on the left axis and proteins on the right one. This is the direct plot of the data obtained from the output file. The data used in this plot represents a simulation output result with the parameter values listed in *Table 14* with 1 splicing site (the values of the *steady state* with 7 splicing sites are similar).

In Figure *7* we can observe the evolution of all the three plotted species during the simulation time. The pre-mRNA reaches a *steady state* of approximately 15 molecules, mRNA *stabilizes* around 23 molecules in close agreement with literature values, indicating an average of 25 (Futcher, Latter, Monardo, McLaughlin, & Garrels). Protein reaches near 15.000 molecules per gene, in aproximately the same order of magnitude reported by (Futcher, Latter, Monardo, McLaughlin, & Garrels), that in yeast observed around 9000 proteins per gene.

3.1 Transcription Initiation

The control within the transcription machinery is shaped by the complex and highly crowded environment in the nucleous cell. Regarding *Figure 8* the transcription initiation rate was varied from 90% to 110% (x axis). The values presented represent the variation of the pre-mRNA, mRNA and protein at A, B and C plots respectively.

In all the three plots is possible to observe the direct correlation between the transcription initiation and the abundance of each one of the species. With the positive slope value, the increase of the initiation rate translates into the correspondent growth on the number of pre-mRNA, mRNA and protein. This behaviour goes toward to what was expected as the increase of the transcription initiation led to the raise of the number of molecules in all the three species (Latchman, 2010).

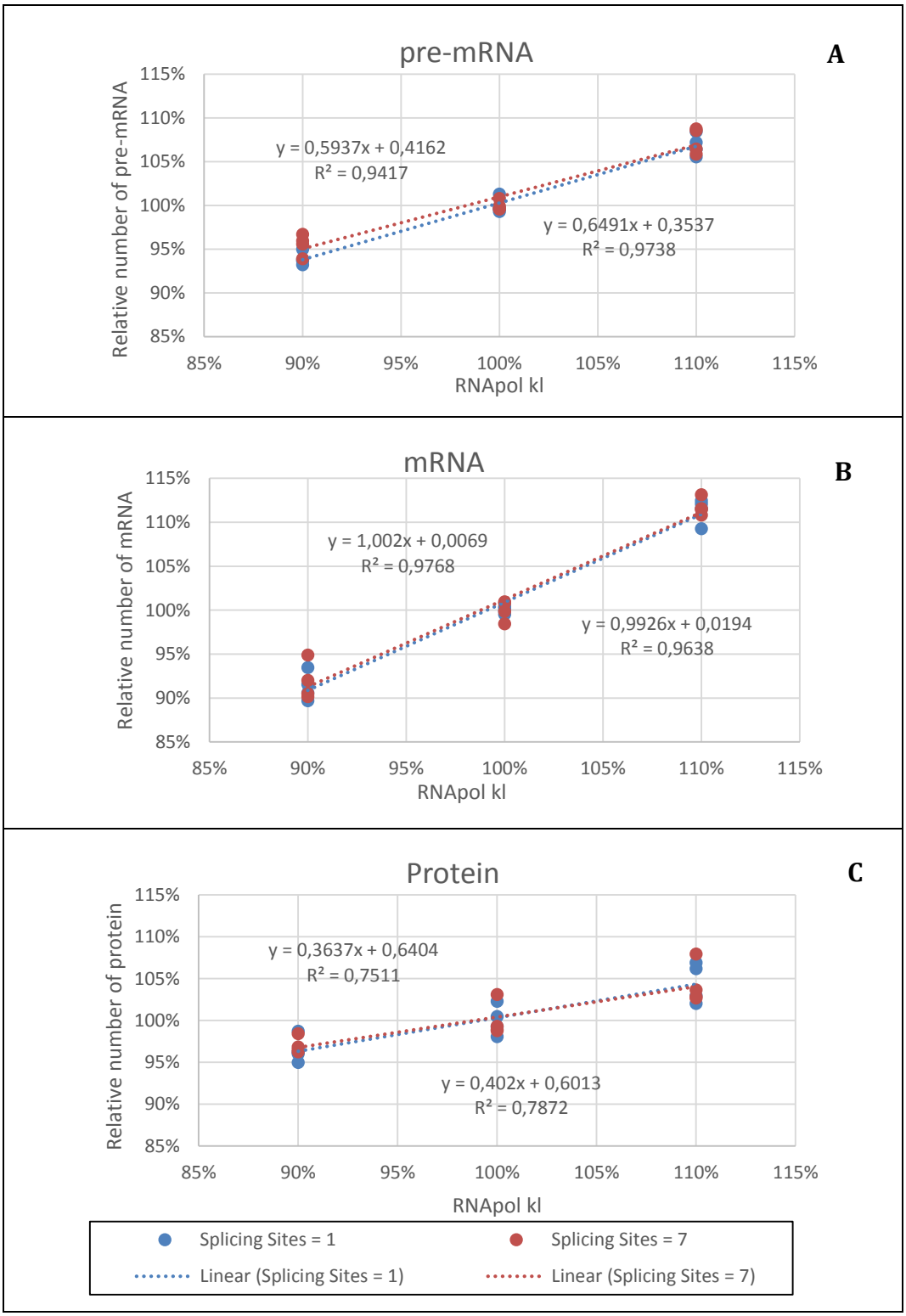

Figure 8: Plot representation of the linear regressions with the variation of the transcription initiation with 100 ribosomes. **A:** Plot of the medium number of pre-mRNA for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red). **B:** Plot of the medium number of mRNA for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red). **C:** Plot of the medium number of protein for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red).

3.2 RNA degradation

General discard pathways eliminate unprocessed and irregular pre-mRNAs to control the quality of gene expression. In contrast to such general pre-mRNA decay, other pathways control the expression of select intron-containing genes (Lemieux, et al., 2011). Here we simulate the general degradation of all the RNAs.

Looking to *Figure 9* the RNA degradation rate was varied between 90% and 110% (x axis) the values plotted represent the variation of the pre-mRNA, mRNA and protein at A, B and C plots respectively. In the next three plots is possible to identify a negative correlation between RNA degradation and pre-mRNA, mRNA and protein, respectively. As the RNA degradation augments, the abundance of the species decreases. The negative slope values in all regressions translates the expected behaviour of diminished number of all the three species as the RNA degradation increases.

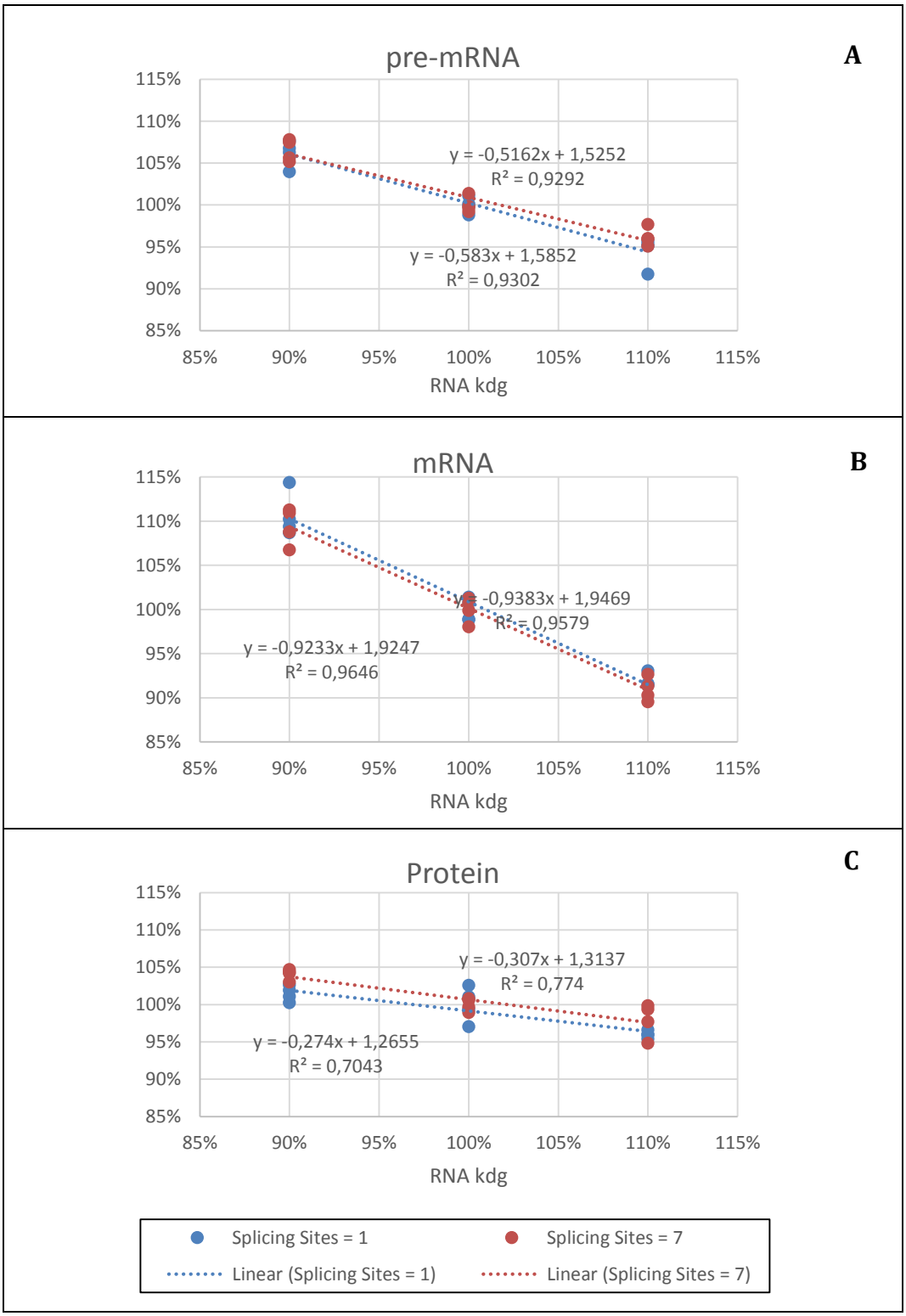

Figure 9: Plot representation of the linear regressions with the variation of the RNA degradation with 100 ribosomes. **A:** Plot of the medium number of pre-mRNA for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red). **B:** Plot of the medium number of mRNA for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red). **C:** Plot of the medium number of protein for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red).

3.3 Protein degradation

The levels of proteins within cells are determined not only by rates of synthesis, but also by rates of degradation. The half-lives of proteins within cells vary widely, from minutes to several days, and differential rates of protein degradation are an important aspect of cell regulation (Cooper, 2000).

In *Figure 10* the plots represent the data referent to the variation of the protein degradation rate, between 90% and 110% (x axis), pre-mRNA, mRNA and protein at A, B and C plots respectively. The number of pre-mRNA and mRNA (A e B) show no response to the protein degradation rate change. As would be expected, the parameter change will affect directly the final stage of gene expression – the protein stability. Therefore the only change should be on the protein number. Looking to the protein plot (C) it is possible to perceive the negative correlation between protein degradation and protein number. With the protein degradation constant increase the number of proteins decreases.

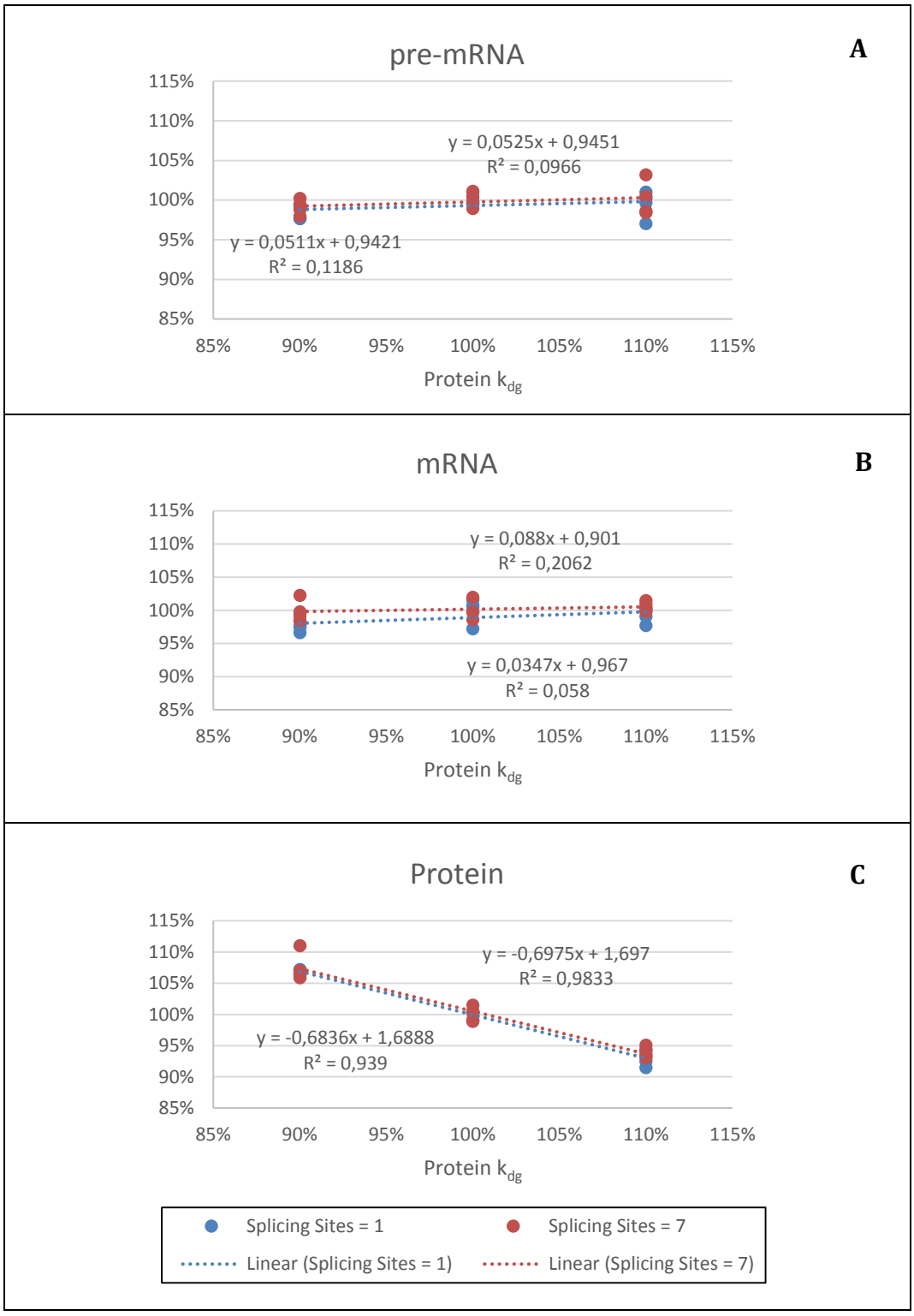

Figure 10: Plot representation of the linear regressions with the variation of the protein degradation with 100 ribosomes. **A:** Plot of the medium number of pre-mRNA for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red). **B:** Plot of the medium number of mRNA for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red). **C:** Plot of the medium number of protein for each one of the 24 simulations (12 with one splicing site in blue and 12 with 7 splicing sites in red).

Looking back at *Figure 8*, *9* and *10* each plot represents both the regressions from the simulations with 1 and 7 splice sites. Comparing them (1 vs. 7) we can observe that the number of introns does not affect the impact of the parameters on the species number. The number of introns defines the magnitude (greater or lower) of the delay in the process of gene expression, but does not affect the degree of control of any of the three steps in the overall velocity of the gene expression.

3.4 Parameters Sensitivity

With the normalized data used to design this plots, it is possible to compare directly the slopes between groups of simulations – transcription initiation, RNA degradation and protein degradation and identify which one of the stages has the major effect on the change of the number of the species.

The following plots represent the absolute value of the slopes so they are comparable between each other in order to evaluate their impact on each specie, regardless of the effective increase or decrease in the molecules (pre-mRNA, mRNA and protein) number.

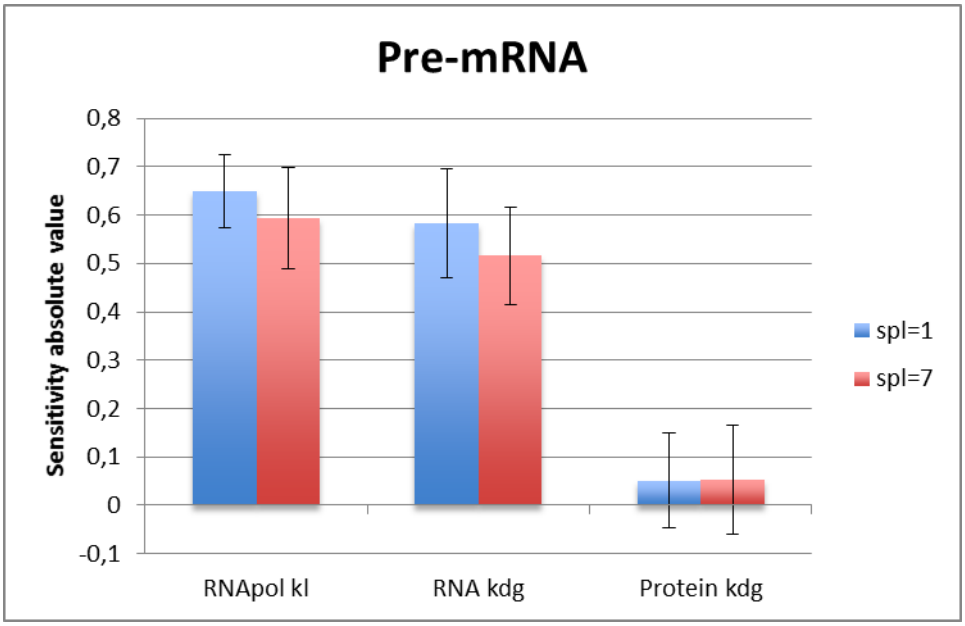

Figure 11: Sensitivity absolute values, obtained from the linear regression slopes. Error bars represent 95% confidence intervals of the pre-mRNA sensitivity (in absolute value) to each parameter (RNApol ki - transcription initiation rate constant, RNA kdg - mRNA degradation rate constant, Protein kdg - protein degradation rate constant).

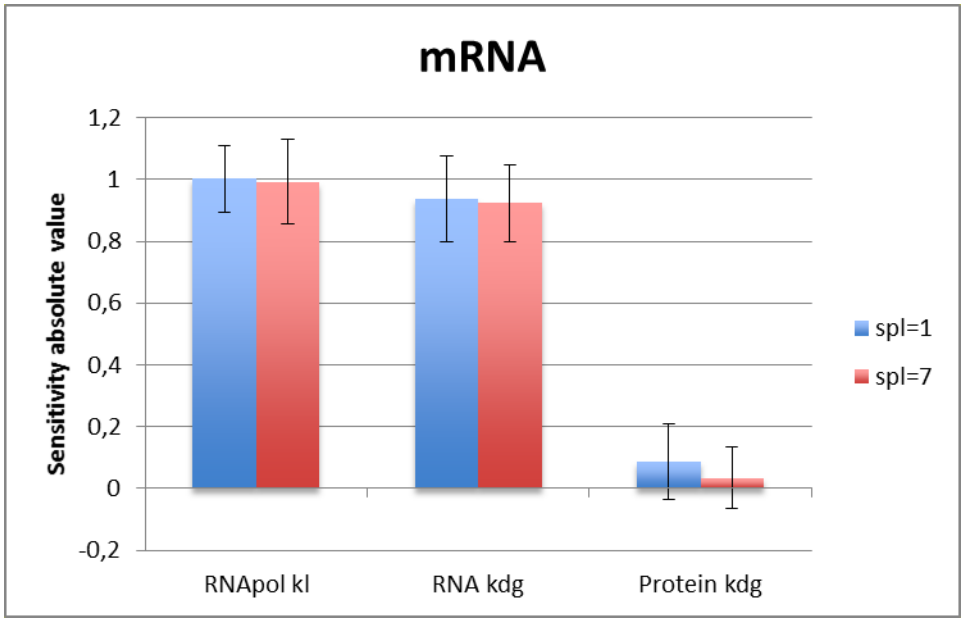

Figure 12: Sensitivity absolute values, obtained from the linear regression slopes. Error bars represent 95% confidence intervals of the mRNA sensitivity (in absolute value) to each parameter (RNApol ki - transcription initiation rate constant, RNA kdg - mRNA degradation rate constant, Protein kdg - protein degradation rate constant).

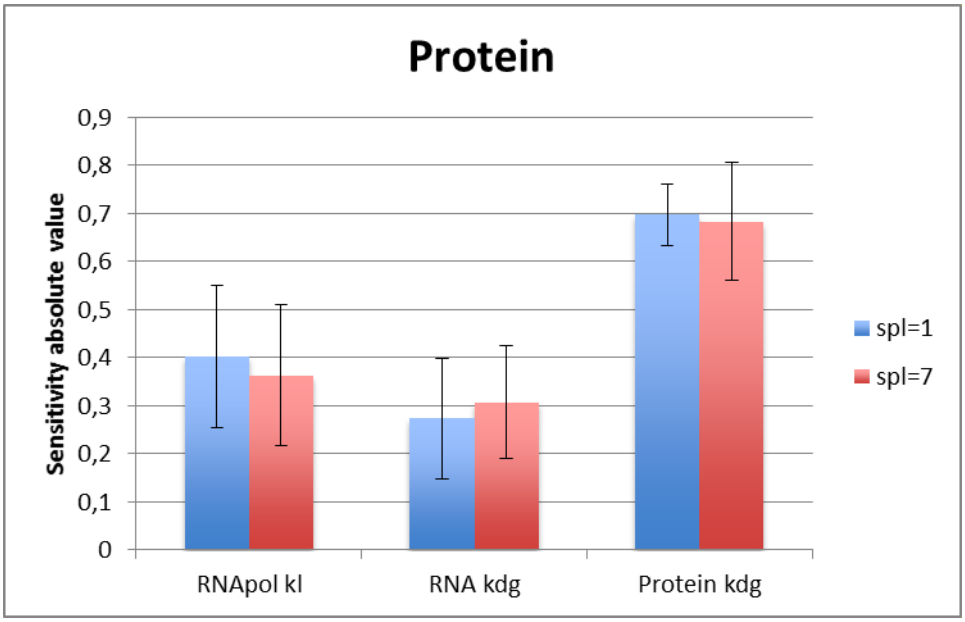

Figure 13: Sensitivity absolute values, obtained from the linear regression slopes. Error bars represent 95% confidence intervals of the protein sensitivity (in absolute value) to each parameter (RNApol ki - transcription initiation rate constant, RNA kdg - mRNA degradation rate constant, Protein kdg - protein degradation rate constant).

By the observation of the *Figure 11* and *Figure 12* we can verify that the protein degradation has no effect on the pre-mRNA and mRNA, statistically verified by the confidence intervals that include zero.

Looking to all the three plots (*Figure 11*, *Figure 12* and *Figure 13*) the initiation constant appears to have a slightly greater impact than RNA degradation in all of them,

but comparing the confidence intervals there is a great overlapping that suggests there is no significant difference.

In *Figure 13* is possible to see a clear difference between the transcription initiation and RNA degradation *versus* the protein degradation. According to the absolute value of the slopes this suggests that the protein degradation has a greater impact on protein number than the transcription initiation and RNA degradation.

3.5 Final Conclusions

We present a Java stochastic model of gene expression including splicing events that can perform simulations of great dimension (15.000 s and up to 50 million of iterations) under 120 minutes. Our method is relatively computationally intensive but it possible to run simultaneously several simulations (for instance all the replicates at once) as long as the machine used has available processors. It is a light program, around 30KB and due to the language used, which runs on a virtually machine, it can be executed on any operating system.

The output data is concise and easy to manipulate by other technologies for the desired outcome. Though the data collected here were only the simulation time and the pre-mRNA, mRNA and protein number, the design of the source code allows it to be easily converted to give other type of output. For example the number of transcripts of pre-mRNA synthesized per RNApol, protein per ribosome, among others.

Although several other methods have been described (Mäkelä, Lloyde-Price, Yli-Harja, & Ribeiro, 2011) (Ribeiro, Häkkinen, & Lloyd-Price, 2012), all of those have been design as an approximation to the bacterial gene expression process. With this model, including the process of splicing with the possibility of varying the number of splicing sites, we aim to take a step closer to the *eukaryotic* transcription process. This allows considerably greater accuracy in the analysis, being the splicing an essential step of the gene expression.

The executed simulations show that our model gives predictions on molecules number within the experimentally observed range. The parameters perturbations conducted to logically expected outcomes of the abundance of all the three species. With this we conclude that our model has proven to produce credible simulations and behaviours. The more detailed conclusions about the relative influence of transcription initiation, RNA degradation and protein degradation influence on gene expression and the impact of introns number are valid for the group of parameters tested, and these can

vary from an organism to other. Moreover, many of the parameters are based on data retrieved from prokaryotes. In the future, a more comprehensive study should be done, one that includes a larger number of parameters to be tested. Particularly, the number of RNA polymerases, spliceosomes and ribosomes, since these, when in small number, can introduce bottlenecks creating a significant impact on the control of gene expression.

Bibliography

Berget, S. M. (1977). Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences*, (74)3171.

Clancy, S. (2008). Introns, Exons and Spliceosome. *Nature Education*, 1(1), 31.

Cooper, G. M. (2000). *The Cell.* Boston: Sinauer Associates, Inc.

Epshtein, V., & Nudler, E. (2003). Cooperation Between RNA Polymerase Molecules in Transcription Elongation. *SCIENCE*, 801-805.

Firczuk, H., Kannambath, S., Pahle, J., Claydon, A., Beynon, Duncan, J., McCarthy, J. E. (2013). An in vivo control map for the eukaryotic mRNA translation machinery. *Molecular Systems Biology*, 9:635.

Futcher, B., Latter, G. I., Monardo, P., McLaughlin, C. S., & Garrels, J. I. (1999). A Sampling of the Yeast Proteome. *Molecular and Cellular Biology*, 7357–7368.

Gillespie, D. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics, 22*(4).

Gillespie, D. (1977). Exact Stochastic Simulation of Coupled Chemical Reactions. *The Journal of Physical Chemistry*, 81(25), 2340.

Greive, S. J., & Hippel, P. H. (2005). Thinking quantitatively about transcriptional regulation. *Nature Reviews*, 221-232.

Greive, S. J., Weitzel, S. E., Goodarzi, J. P., Main, L. J., Pasman, Z., & Hippel, P. H. (2008). Monitoring RNA transcription in real time by using surface plasmon resonance. *PNAS*, 3315–3320.

Grünwald, D., Singer, R. H., & Rout, M. (2011). Nuclear export dynamics of RNA-protein complexes. *Nature*, 333-341.

Huranová, M., Ivani, I., Benda, A., Poser, I., Brody, Y., Hof, M., Stanek, D. (2010). The differential interaction of snRNPs with pre-mRNA reveals splicing kinetics in living cells. *The Journal of Cell Biology*, 75-86.

Keren, H., Lev-Maor, G., & Ast, G. (2010). Alternative splicing and evolution: diversification, exon definition and function. *Nature Reviews*, (11)345.

Krebs, J. E., & Lewin, B. (2011). *Lewin's Genes X.* Jones and Bartlett.

Latchman, D. S. (2010). *Gene Control.* New York: Garland Science.

Lemieux, C., Marguerat, S., Lafontaine, J., Barbezier, N., Bähler, J., & Bachand, F. (2011). A Pre-mRNA Degradation Pathway that Selectively Targets Intron-Containing Genes Requires the Nuclear Poly(A)-Binding Protein. *Molecular Cell*, 108–119.

Mäkelä, J., Lloyde-Price, J., Yli-Harja, O., & Ribeiro, A. S. (2011). Stochastic sequence-level model of coupled transcription and translation in prokaryotes. *BMC Bioinformatics*.

Nelson, D. L., & Cox, M. M. (2008). *Lehninger Principles of Biochemistry.* New York: W. H. Freeman and Company.

Oracle Corporation. (2014). *Java Documentation*. Retrieved from The Java Tutorials - OOP Concepts: https://docs.oracle.com/javase/tutorial/java/concepts/

Oracle Corporation. (2014). *Java Documentation*. Retrieved from The Java Tutorials - Primitive Data Types: https://docs.oracle.com/javase/java/nutsandbolts/datatypes.html

Oracle Corporation. (2014). *Java Documentation*. Retrieved from The Java Tutorials - Classes and Objects: https://docs.oracle.com/javase/tutorial/java/javaOO/index.html

Oracle Corporation. (2014). *Java Documentation*. Retrieved from The Java Tutorials - Controlling Access to Members of a Class: https://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html

Oracle Corporation. (2014). *Learn About Java Technology*. Retrieved from Java Website: www.java.com/en/about

Oracle Corporation. (2014). *The History of Java Technology*. Retrieved from Oracle website: http://www.oracle.com

Potapov, I., Mäkela, J., Yli-Harja, O., & Ribeiro, A. S. (2012). Effects of codon sequence on the dynamics of genetic networks. *Journal of Theoretical Biology*, 17-25.

Rao, C. V., & Arkin, A. P. (2003). Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *Journal of Chemical Physics*, 18(11), 4999.

Ribeiro, A. S., Häkkinen, A., & Lloyd-Price, J. (2012). Effects of gene length on the dynamics of gene expression. *Computational Biology and Chemistry*, 1-9.

Roussel, M. R. (2006). Validation of an algorithm for delay stochastic simulation of transcription and translation in prokariotic gene expression. *Physical Biology*, 3(4), 274.

Singh, J., & Padgett, R. A. (2009). Rates of in situ transcription and splicing in large human genes. *Nature Structural & Molecular Biology*, 1128-1133.

Stirzaker, D. (2005). *Stochastic Processes & Models.* New York: Oxford University Press.

Wang, E. T. (2010). Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456-470.

White, R. J. (2001). *Gene Transcription, Mechanisms and Control.* Glasgow, UK: Blackwell Science.

Will, C. L., & Lührmann, R. (2011). Spliceosome Structure and Function. *Cold Spring Harbor Perspectives in Biology*, 3:a003707.

Yu, J. e. (2006). Probing gene expression in live cells, one protein molecule at a time. *Science*, 311(5767), 1600.

Zhu, R., Ribeiro, A. S., Salahub, D., & Kauffman, S. A. (2007). Studying genetic regulatory networks at the molecular level: Delayed reaction stochastic models. *Journal of Theoretical Biology*, 725-745.

Appendix

UML (Unified Modelling Language) representation of the model classes and their relations.